Deep Learning Model

Model Pseudocode

In [2]:

```
# Function create_cnn_model(input_shape):

#     Initialize Sequential model

#     Add Conv1D(32, kernel=3, padding='same', input_shape)
#     Add BatchNormalization
#     Add LeakyReLU(α=0.3)
#     Add MaxPooling1D(pool=2)

#     Add Conv1D(64, kernel=3, padding='same')
#     Add BatchNormalization
#     Add LeakyReLU(α=0.3)
#     Add MaxPooling1D(pool=2)

#     Add Conv1D(128, kernel=3, padding='same')
#     Add BatchNormalization
#     Add LeakyReLU(α=0.3)

#     Add GlobalAveragePooling1D

#     Add Dense(64) → BatchNormalization → LeakyReLU(α=0.3)
#     Add Dropout(0.25)

#     Add Output Dense(1, activation='sigmoid')

#     Compile model with:
#         optimizer = 'adam'
#         loss = 'binary_crossentropy'
#         metrics = ['accuracy']

#     Return model
```

Step 1: Load and Prepare the Dataset

In [3]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import MinMaxScaler

# Load dataset
df = pd.read_csv('D:\Coding Projects\Detection-of-SYN-Flood-Attacks-Using-Machine-L
X = df.drop('Label', axis=1).values
y = df['Label'].values
X = X.reshape(X.shape[0], X.shape[1], 1)

# Flatten before scaling and reshape after
X_flat = X.reshape(X.shape[0], X.shape[1])
```

```
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X_flat)

# Reshape back to 3D for CNN input
X = X_scaled.reshape(X.shape[0], X.shape[1], 1)
```

Step 2: Defining the 1D CNN Architecture

In [4]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, GlobalAveragePooling1D, D

def CCN_Model(input_shape):
    model = Sequential()

    model.add(Conv1D(32, kernel_size=3, padding='same', input_shape=input_shape))
    model.add(BatchNormalization())
    model.add(LeakyReLU(alpha=0.3))
    model.add(MaxPooling1D(pool_size=2))

    model.add(Conv1D(64, kernel_size=3, padding='same'))
    model.add(BatchNormalization())
    model.add(LeakyReLU(alpha=0.3))
    model.add(MaxPooling1D(pool_size=2))

    model.add(Conv1D(128, kernel_size=3, padding='same'))
    model.add(BatchNormalization())
    model.add(LeakyReLU(alpha=0.3))

    model.add(GlobalAveragePooling1D())

    model.add(Dense(64))
    model.add(BatchNormalization())
    model.add(LeakyReLU(alpha=0.3))
    model.add(Dropout(0.25))  # Moderate regularization

    model.add(Dense(1, activation='sigmoid'))

    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'
    return model
```

Step 3: Training the Model using the K-Folds + Resource Management

In [ ]:

```
import time
import psutil
import os

accuracies = []
true_label = []
pred_label = []
fold_results = []

process = psutil.Process(os.getpid())
```

```
# Resource Monitoring Start
start_time_count = time.time()
start_ram_count = process.memory_info().rss / 1024 / 1024  # in MB
start_cpu_count = psutil.cpu_percent(interval=1)

for fold in range(0, 5):
    print(f"\n--- Training on Fold {fold} ---")

    train_idx = df[df['Fold'] != fold].index
    test_idx = df[df['Fold'] == fold].index

    X_train, X_test = X[train_idx], X[test_idx]
    y_train, y_test = y[train_idx], y[test_idx]

    model = CCN_Model(input_shape=X.shape[1:])

    # Train Model
    history = model.fit(
        X_train, y_train,
        epochs=30,
        batch_size=64,
        validation_data=(X_test, y_test),
        verbose=1
    )

    #  Evaluation
    y_scores = model.predict(X_test).ravel()
    y_pred = (y_scores > 0.5).astype(int)

    true_label.extend(y_test)
    pred_label.extend(y_pred)
    fold_results.extend(y_scores)

    loss, acc = model.evaluate(X_test, y_test, verbose=0)
    accuracies.append(acc)

# Resource Monitoring End
end_time_count = time.time()
end_ram_count = process.memory_info().rss / 1024 / 1024  # in MB
end_cpu_count = psutil.cpu_percent(interval=1)

# Summary
print("\n Overall Training Stats ")
print(f"Total Training Time: {end_time_count - start_time_count:.2f} seconds")
print(f"Total RAM Usage Increase: {end_ram_count - start_ram_count:.2f} MB")
print(f"CPU Usage (at final check): {end_cpu_count}%")
```

```
--- Training on Fold 0 ---
Epoch 1/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.9879 - loss: 0.0622 - val_acc
uracy: 0.5154 - val_loss: 0.9026
Epoch 2/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9968 - loss: 0.0165 - val_acc
uracy: 0.5154 - val_loss: 0.6828
Epoch 3/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9965 - loss: 0.0158 - val_acc
uracy: 1.0000 - val_loss: 0.1328
Epoch 4/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9966 - loss: 0.0125 - val_acc
uracy: 1.0000 - val_loss: 0.0092
Epoch 5/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9979 - loss: 0.0101 - val_acc
uracy: 0.9995 - val_loss: 0.0095
Epoch 6/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9989 - loss: 0.0074 - val_acc
uracy: 0.9995 - val_loss: 0.0031
Epoch 7/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9977 - loss: 0.0126 - val_acc
uracy: 0.9995 - val_loss: 0.0062
Epoch 8/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9975 - loss: 0.0109 - val_acc
uracy: 0.9896 - val_loss: 0.1405
Epoch 9/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9945 - loss: 0.0261 - val_acc
uracy: 0.9995 - val_loss: 0.0041
Epoch 10/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0090 - val_acc
uracy: 0.9984 - val_loss: 0.0037
Epoch 11/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9964 - loss: 0.0141 - val_acc
uracy: 0.9995 - val_loss: 0.0026
Epoch 12/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9964 - loss: 0.0114 - val_acc
uracy: 0.9995 - val_loss: 0.0299
Epoch 13/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9925 - loss: 0.0359 - val_acc
uracy: 0.9990 - val_loss: 0.0060
Epoch 14/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9972 - loss: 0.0105 - val_acc
uracy: 0.9990 - val_loss: 0.0039
Epoch 15/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9956 - loss: 0.0177 - val_acc
uracy: 1.0000 - val_loss: 0.0029
Epoch 16/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9985 - loss: 0.0091 - val_acc
uracy: 0.9984 - val_loss: 0.0041
Epoch 17/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9979 - loss: 0.0127 - val_acc
uracy: 0.9984 - val_loss: 0.0042
Epoch 18/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9982 - loss: 0.0099 - val_acc
uracy: 0.9995 - val_loss: 0.0023
Epoch 19/30
```

```
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9976 - loss: 0.0081 - val_acc
uracy: 0.9995 - val_loss: 0.0018
Epoch 20/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9983 - loss: 0.0075 - val_acc
uracy: 1.0000 - val_loss: 0.0013
Epoch 21/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9990 - loss: 0.0061 - val_acc
uracy: 1.0000 - val_loss: 0.0018
Epoch 22/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9978 - loss: 0.0105 - val_acc
uracy: 1.0000 - val_loss: 0.0016
Epoch 23/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9986 - loss: 0.0082 - val_acc
uracy: 1.0000 - val_loss: 0.0012
Epoch 24/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9983 - loss: 0.0070 - val_acc
uracy: 0.9990 - val_loss: 0.0016
Epoch 25/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9985 - loss: 0.0054 - val_acc
uracy: 0.9995 - val_loss: 0.0025
Epoch 26/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9977 - loss: 0.0092 - val_acc
uracy: 0.9974 - val_loss: 0.0046
Epoch 27/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9994 - loss: 0.0059 - val_acc
uracy: 0.9995 - val_loss: 0.0022
Epoch 28/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9986 - loss: 0.0057 - val_acc
uracy: 1.0000 - val_loss: 0.0033
Epoch 29/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9889 - loss: 0.0387 - val_acc
uracy: 0.9990 - val_loss: 0.0017
Epoch 30/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9974 - loss: 0.0125 - val_acc
uracy: 1.0000 - val_loss: 0.0030
61/61 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step

--- Training on Fold 1 ---
Epoch 1/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.9757 - loss: 0.0753 - val_acc
uracy: 0.5154 - val_loss: 1.3911
Epoch 2/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9973 - loss: 0.0178 - val_acc
uracy: 0.5154 - val_loss: 1.5026
Epoch 3/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9975 - loss: 0.0134 - val_acc
uracy: 0.5154 - val_loss: 0.5363
Epoch 4/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0082 - val_acc
uracy: 0.9979 - val_loss: 0.0747
Epoch 5/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9986 - loss: 0.0078 - val_acc
uracy: 0.9990 - val_loss: 0.0115
Epoch 6/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9935 - loss: 0.0227 - val_acc
uracy: 0.9990 - val_loss: 0.0054
```

```
Epoch 7/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0093 - val_acc
uracy: 0.9984 - val_loss: 0.0048
Epoch 8/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9986 - loss: 0.0072 - val_acc
uracy: 0.9995 - val_loss: 0.0021
Epoch 9/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9987 - loss: 0.0077 - val_acc
uracy: 0.9984 - val_loss: 0.0091
Epoch 10/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9991 - loss: 0.0062 - val_acc
uracy: 0.9990 - val_loss: 0.0030
Epoch 11/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9898 - loss: 0.0255 - val_acc
uracy: 0.9995 - val_loss: 0.0028
Epoch 12/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9985 - loss: 0.0097 - val_acc
uracy: 0.9984 - val_loss: 0.0048
Epoch 13/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9970 - loss: 0.0125 - val_acc
uracy: 0.9995 - val_loss: 0.0022
Epoch 14/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9991 - loss: 0.0043 - val_acc
uracy: 0.9995 - val_loss: 0.0026
Epoch 15/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0150 - val_acc
uracy: 0.9995 - val_loss: 0.0028
Epoch 16/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9982 - loss: 0.0096 - val_acc
uracy: 0.9995 - val_loss: 0.0023
Epoch 17/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9977 - loss: 0.0082 - val_acc
uracy: 0.9995 - val_loss: 0.0021
Epoch 18/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0098 - val_acc
uracy: 0.9990 - val_loss: 0.0023
Epoch 19/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9989 - loss: 0.0056 - val_acc
uracy: 0.9995 - val_loss: 0.0018
Epoch 20/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9978 - loss: 0.0078 - val_acc
uracy: 0.9995 - val_loss: 0.0019
Epoch 21/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9984 - loss: 0.0080 - val_acc
uracy: 0.9995 - val_loss: 0.0020
Epoch 22/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0078 - val_acc
uracy: 0.9995 - val_loss: 0.0016
Epoch 23/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0085 - val_acc
uracy: 0.9995 - val_loss: 0.0017
Epoch 24/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9984 - loss: 0.0058 - val_acc
uracy: 0.9984 - val_loss: 0.0097
Epoch 25/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9978 - loss: 0.0083 - val_acc
```

```
uracy: 0.8006 - val_loss: 0.1918
Epoch 26/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9869 - loss: 0.0347 - val_acc
uracy: 0.9995 - val_loss: 0.0022
Epoch 27/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0064 - val_acc
uracy: 0.9984 - val_loss: 0.0043
Epoch 28/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9972 - loss: 0.0121 - val_acc
uracy: 0.9974 - val_loss: 0.0199
Epoch 29/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9953 - loss: 0.0196 - val_acc
uracy: 0.9974 - val_loss: 0.0213
Epoch 30/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9966 - loss: 0.0135 - val_acc
uracy: 0.9995 - val_loss: 0.0017
61/61 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step

--- Training on Fold 2 ---
Epoch 1/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.9699 - loss: 0.0862 - val_acc
uracy: 0.5154 - val_loss: 1.1742
Epoch 2/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9986 - loss: 0.0094 - val_acc
uracy: 0.5154 - val_loss: 1.3882
Epoch 3/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0076 - val_acc
uracy: 0.9948 - val_loss: 0.2058
Epoch 4/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9982 - loss: 0.0091 - val_acc
uracy: 0.9932 - val_loss: 0.2444
Epoch 5/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9971 - loss: 0.0190 - val_acc
uracy: 0.9958 - val_loss: 0.0175
Epoch 6/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9984 - loss: 0.0071 - val_acc
uracy: 0.9974 - val_loss: 0.0157
Epoch 7/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0075 - val_acc
uracy: 0.9958 - val_loss: 0.0174
Epoch 8/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9950 - loss: 0.0181 - val_acc
uracy: 0.9969 - val_loss: 0.0161
Epoch 9/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9989 - loss: 0.0059 - val_acc
uracy: 0.9969 - val_loss: 0.0159
Epoch 10/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9980 - loss: 0.0070 - val_acc
uracy: 0.9974 - val_loss: 0.0142
Epoch 11/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9986 - loss: 0.0082 - val_acc
uracy: 0.9974 - val_loss: 0.0139
Epoch 12/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9991 - loss: 0.0054 - val_acc
uracy: 0.9969 - val_loss: 0.0149
Epoch 13/30
```

```
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9982 - loss: 0.0113 - val_acc
uracy: 0.9974 - val_loss: 0.0136
Epoch 14/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9992 - loss: 0.0043 - val_acc
uracy: 0.9979 - val_loss: 0.0168
Epoch 15/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9940 - loss: 0.0280 - val_acc
uracy: 0.9974 - val_loss: 0.0146
Epoch 16/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9949 - loss: 0.0200 - val_acc
uracy: 0.9974 - val_loss: 0.0135
Epoch 17/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9986 - loss: 0.0090 - val_acc
uracy: 0.9974 - val_loss: 0.0137
Epoch 18/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9978 - loss: 0.0091 - val_acc
uracy: 0.9958 - val_loss: 0.0150
Epoch 19/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9987 - loss: 0.0053 - val_acc
uracy: 0.9974 - val_loss: 0.0136
Epoch 20/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0073 - val_acc
uracy: 0.9974 - val_loss: 0.0135
Epoch 21/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9990 - loss: 0.0050 - val_acc
uracy: 0.9964 - val_loss: 0.0134
Epoch 22/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9995 - loss: 0.0030 - val_acc
uracy: 0.9964 - val_loss: 0.0151
Epoch 23/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9994 - loss: 0.0040 - val_acc
uracy: 0.9979 - val_loss: 0.0124
Epoch 24/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9986 - loss: 0.0053 - val_acc
uracy: 0.9974 - val_loss: 0.0141
Epoch 25/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0053 - val_acc
uracy: 0.9958 - val_loss: 0.0136
Epoch 26/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0066 - val_acc
uracy: 0.9974 - val_loss: 0.0124
Epoch 27/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0046 - val_acc
uracy: 0.9969 - val_loss: 0.0131
Epoch 28/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9993 - loss: 0.0045 - val_acc
uracy: 0.9964 - val_loss: 0.0167
Epoch 29/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9989 - loss: 0.0056 - val_acc
uracy: 0.9979 - val_loss: 0.0133
Epoch 30/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9994 - loss: 0.0032 - val_acc
uracy: 0.9979 - val_loss: 0.0128
61/61 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step

--- Training on Fold 3 ---
```

```
Epoch 1/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.9633 - loss: 0.0907 - val_acc
uracy: 0.5154 - val_loss: 1.1037
Epoch 2/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0091 - val_acc
uracy: 0.5154 - val_loss: 1.3199
Epoch 3/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9986 - loss: 0.0065 - val_acc
uracy: 0.7439 - val_loss: 0.3882
Epoch 4/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0047 - val_acc
uracy: 0.9969 - val_loss: 0.0244
Epoch 5/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9973 - loss: 0.0178 - val_acc
uracy: 0.9969 - val_loss: 0.0206
Epoch 6/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9984 - loss: 0.0067 - val_acc
uracy: 0.9969 - val_loss: 0.0242
Epoch 7/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9989 - loss: 0.0081 - val_acc
uracy: 0.9979 - val_loss: 0.0195
Epoch 8/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9992 - loss: 0.0046 - val_acc
uracy: 0.9979 - val_loss: 0.0173
Epoch 9/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9990 - loss: 0.0031 - val_acc
uracy: 0.9974 - val_loss: 0.0213
Epoch 10/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9935 - loss: 0.0194 - val_acc
uracy: 0.9964 - val_loss: 0.0185
Epoch 11/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9985 - loss: 0.0085 - val_acc
uracy: 0.9964 - val_loss: 0.0194
Epoch 12/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9987 - loss: 0.0059 - val_acc
uracy: 0.9969 - val_loss: 0.0243
Epoch 13/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9978 - loss: 0.0160 - val_acc
uracy: 0.9979 - val_loss: 0.0179
Epoch 14/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0053 - val_acc
uracy: 0.9969 - val_loss: 0.0254
Epoch 15/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9987 - loss: 0.0127 - val_acc
uracy: 0.9974 - val_loss: 0.0202
Epoch 16/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9978 - loss: 0.0092 - val_acc
uracy: 0.9969 - val_loss: 0.0214
Epoch 17/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9979 - loss: 0.0096 - val_acc
uracy: 0.9979 - val_loss: 0.0184
Epoch 18/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9982 - loss: 0.0073 - val_acc
uracy: 0.9979 - val_loss: 0.0183
Epoch 19/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9990 - loss: 0.0036 - val_acc
```

```
uracy: 0.9979 - val_loss: 0.0195
Epoch 20/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9987 - loss: 0.0059 - val_acc
uracy: 0.9979 - val_loss: 0.0188
Epoch 21/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9996 - loss: 0.0031 - val_acc
uracy: 0.9979 - val_loss: 0.0197
Epoch 22/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9984 - loss: 0.0053 - val_acc
uracy: 0.9979 - val_loss: 0.0208
Epoch 23/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9996 - loss: 0.0019 - val_acc
uracy: 0.9969 - val_loss: 0.0230
Epoch 24/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9989 - loss: 0.0035 - val_acc
uracy: 0.9979 - val_loss: 0.0212
Epoch 25/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9993 - loss: 0.0023 - val_acc
uracy: 0.9979 - val_loss: 0.0215
Epoch 26/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9990 - loss: 0.0027 - val_acc
uracy: 0.9974 - val_loss: 0.0204
Epoch 27/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9989 - loss: 0.0075 - val_acc
uracy: 0.9938 - val_loss: 0.0269
Epoch 28/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9987 - loss: 0.0059 - val_acc
uracy: 0.9974 - val_loss: 0.0205
Epoch 29/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9993 - loss: 0.0040 - val_acc
uracy: 0.9974 - val_loss: 0.0209
Epoch 30/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9990 - loss: 0.0026 - val_acc
uracy: 0.9974 - val_loss: 0.0214
61/61 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step

--- Training on Fold 4 ---
Epoch 1/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.9619 - loss: 0.0907 - val_acc
uracy: 0.5151 - val_loss: 1.0424
Epoch 2/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9960 - loss: 0.0141 - val_acc
uracy: 0.5151 - val_loss: 1.4785
Epoch 3/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9974 - loss: 0.0106 - val_acc
uracy: 0.7375 - val_loss: 0.3943
Epoch 4/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9987 - loss: 0.0083 - val_acc
uracy: 0.9974 - val_loss: 0.0910
Epoch 5/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9983 - loss: 0.0084 - val_acc
uracy: 0.9974 - val_loss: 0.0117
Epoch 6/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9983 - loss: 0.0075 - val_acc
uracy: 0.9932 - val_loss: 0.0124
Epoch 7/30
```

```
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9960 - loss: 0.0203 - val_acc
uracy: 0.9932 - val_loss: 0.0373
Epoch 8/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9960 - loss: 0.0254 - val_acc
uracy: 0.9979 - val_loss: 0.0182
Epoch 9/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9992 - loss: 0.0070 - val_acc
uracy: 0.9979 - val_loss: 0.0130
Epoch 10/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0104 - val_acc
uracy: 0.9979 - val_loss: 0.0145
Epoch 11/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0090 - val_acc
uracy: 0.9979 - val_loss: 0.0105
Epoch 12/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9985 - loss: 0.0068 - val_acc
uracy: 0.9979 - val_loss: 0.0072
Epoch 13/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9965 - loss: 0.0175 - val_acc
uracy: 0.9932 - val_loss: 0.0270
Epoch 14/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9978 - loss: 0.0136 - val_acc
uracy: 0.9932 - val_loss: 0.0157
Epoch 15/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9971 - loss: 0.0114 - val_acc
uracy: 0.9979 - val_loss: 0.0115
Epoch 16/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9986 - loss: 0.0081 - val_acc
uracy: 0.9974 - val_loss: 0.0099
Epoch 17/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9976 - loss: 0.0095 - val_acc
uracy: 0.9979 - val_loss: 0.0095
Epoch 18/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9981 - loss: 0.0117 - val_acc
uracy: 0.9979 - val_loss: 0.0101
Epoch 19/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9989 - loss: 0.0065 - val_acc
uracy: 0.9979 - val_loss: 0.0085
Epoch 20/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9982 - loss: 0.0069 - val_acc
uracy: 0.9979 - val_loss: 0.0073
Epoch 21/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9987 - loss: 0.0069 - val_acc
uracy: 0.9979 - val_loss: 0.0061
Epoch 22/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9991 - loss: 0.0061 - val_acc
uracy: 0.9979 - val_loss: 0.0057
Epoch 23/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9980 - loss: 0.0063 - val_acc
uracy: 0.9979 - val_loss: 0.0061
Epoch 24/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9977 - loss: 0.0095 - val_acc
uracy: 0.9974 - val_loss: 0.0073
Epoch 25/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9983 - loss: 0.0072 - val_acc
uracy: 0.9979 - val_loss: 0.0054
```

```
Epoch 26/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9973 - loss: 0.0107 - val_acc
uracy: 0.9979 - val_loss: 0.0067
Epoch 27/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - loss: 0.0049 - val_acc
uracy: 0.9958 - val_loss: 0.0103
Epoch 28/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9987 - loss: 0.0048 - val_acc
uracy: 0.9979 - val_loss: 0.0057
Epoch 29/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9980 - loss: 0.0071 - val_acc
uracy: 0.9984 - val_loss: 0.0073
Epoch 30/30
121/121 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9949 - loss: 0.0153 - val_acc
uracy: 0.9891 - val_loss: 0.0382
60/60 ━━━━━━━━━━━━━━━━━━━━ 0s 928us/step

 Overall Training Stats
Total Training Time: 60.67 seconds
Total RAM Usage Increase: 146.97 MB
CPU Usage (at final check): 10.2%
```

Step 4: Evaluation

In [ ]:

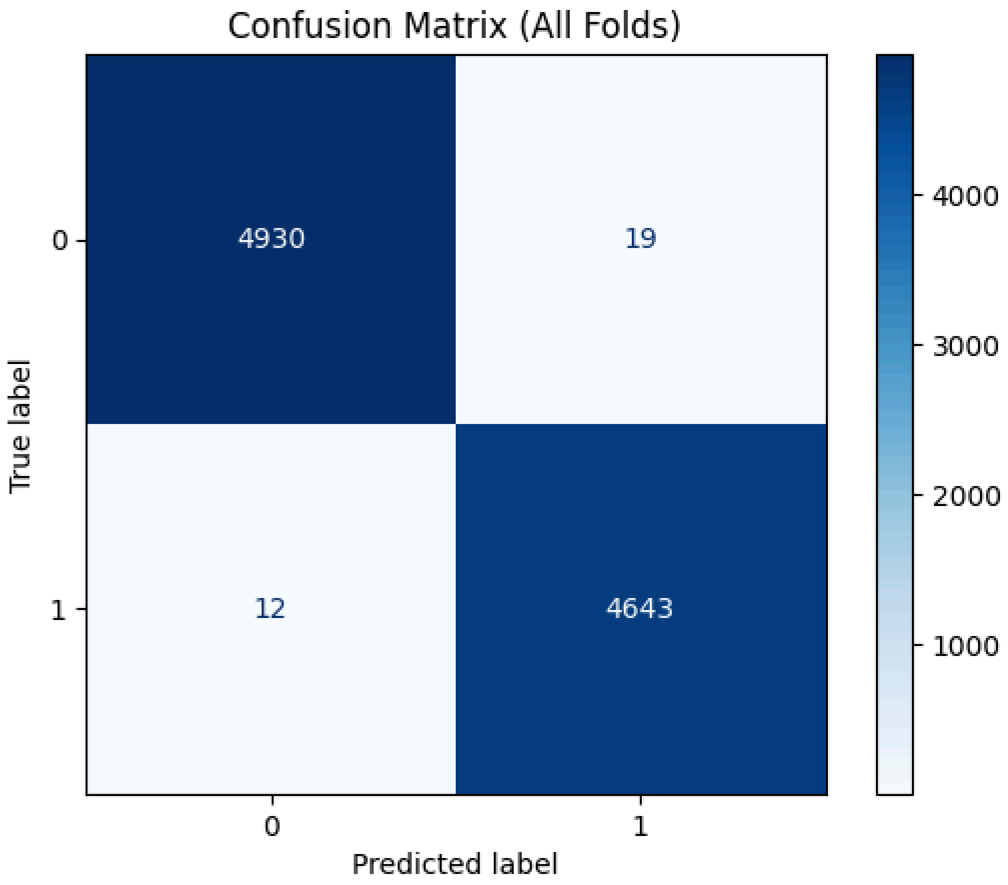
```python
import numpy as np

print("\nFinal CNN Cross-Validation Results:")
print(f"Fold Accuracies: {accuracies}")
print(f"Mean Accuracy: {np.mean(accuracies):.4f}")
print(f"Standard Deviation: {np.std(accuracies):.4f}")
```

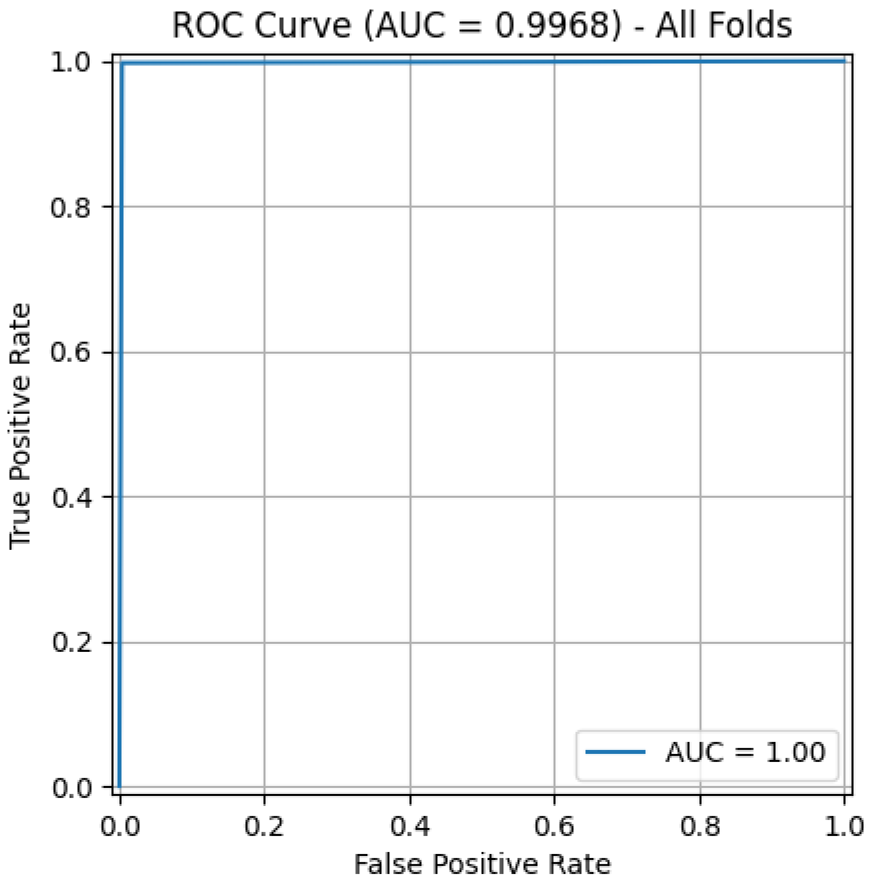
```
Final CNN Cross-Validation Results:
Fold Accuracies: [1.0, 0.9989588856697083, 0.9963560700416565, 0.9963560700416565,
0.7770833373069763]
Mean Accuracy: 0.9538
Standard Deviation: 0.0883
```

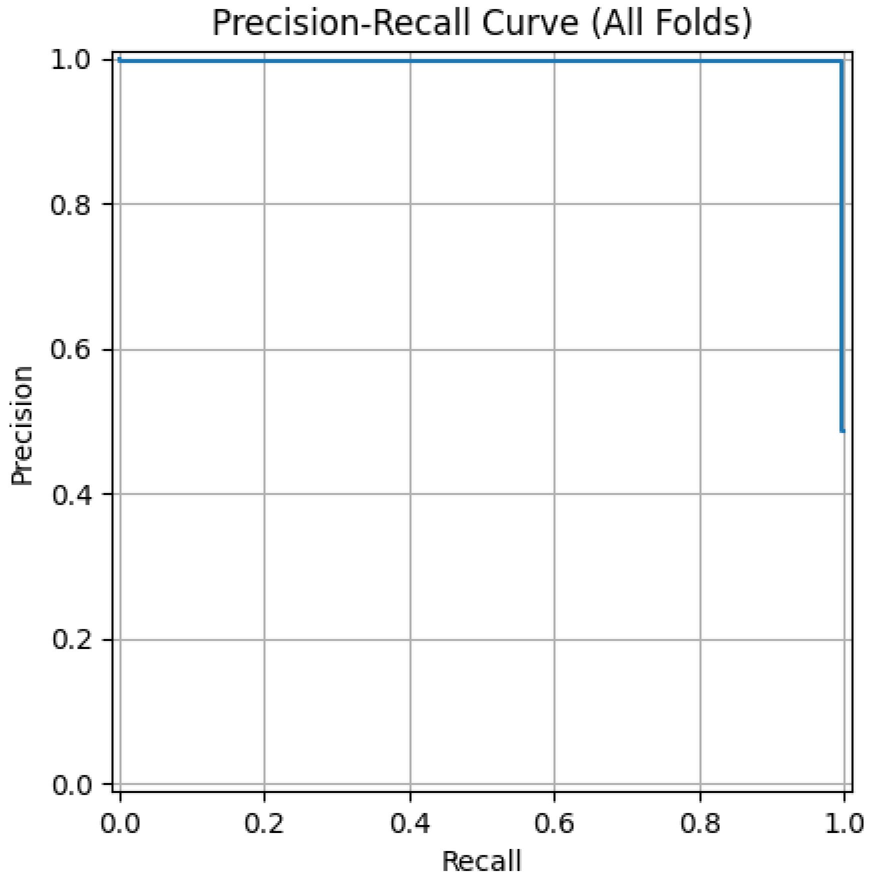
Step 5: Visual Evaluation

In [8]:

```python
import matplotlib.pyplot as plt
from sklearn.metrics import (
    confusion_matrix,
    ConfusionMatrixDisplay,
    roc_curve,
    auc,
    RocCurveDisplay,
    precision_recall_curve,
    PrecisionRecallDisplay
)

# Confusion Matrix
cm = confusion_matrix(true_label, pred_label)
disp_cm = ConfusionMatrixDisplay(confusion_matrix=cm)
disp_cm.plot(cmap='Blues')
plt.title('Confusion Matrix (All Folds)')
plt.grid(False)
plt.show()

# ROC Curve
fpr, tpr, _ = roc_curve(true_label, pred_label)
roc_auc = auc(fpr, tpr)
RocCurveDisplay(fpr=fpr, tpr=tpr, roc_auc=roc_auc).plot()
plt.title(f'ROC Curve (AUC = {roc_auc:.4f}) - All Folds')
plt.grid(True)
plt.show()

# Precision-Recall Curve
precision, recall, _ = precision_recall_curve(true_label, pred_label)
PrecisionRecallDisplay(precision=precision, recall=recall).plot()
plt.title('Precision-Recall Curve (All Folds)')
plt.grid(True)
plt.show()
```




ROC Curve (AUC = 0.9968) - All Folds
True Positive Rate
False Positive Rate
AUC = 1.00
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


Precision-Recall Curve (All Folds)
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

saving the model as PDF

In [ ]:

```
import os
os.getcwd()
```

Out[ ]:

```
'd:\\Coding Projects\\Detection-of-SYN-Flood-Attacks-Using-Machine-Learning-and-De
ep-Learning-Techniques-with-Feature-Base\\Amir Tavahin'
```

In [ ]:

```
!jupyter nbconvert --to webpdf "d:\\Coding Projects\\Detection-of-SYN-Flood-Attacks
```

```
[NbConvertApp] Converting notebook d:\\Coding Projects\\Detection-of-SYN-Flood-Attac
ks-Using-Machine-Learning-and-Deep-Learning-Techniques-with-Feature-Base\\Amir Tavah
in\\CNN.ipynb to webpdf
[NbConvertApp] WARNING | Alternative text is missing on 3 image(s).
[NbConvertApp] Building PDF
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 219122 bytes to d:\Coding Projects\Detection-of-SYN-Flood-Att
acks-Using-Machine-Learning-and-Deep-Learning-Techniques-with-Feature-Base\Amir Tava
hin\CNN.pdf
```